

08/30/00

08/30/00

JCS94 U.S. PTO

08-31-00

A

Please type a plus sign (+) inside this box → ☐PTO/SB/05 (4/98)  
Approved for use through 09/30/2000. OMB 0651-0032  
Patent and Trademark Office: U.S. DEPARTMENT OF COMMERCE

Under the Paperwork Reduction Act of 1995, no persons are required to respond to a collection of information unless it displays a valid OMB control number.

**UTILITY  
PATENT APPLICATION  
TRANSMITTAL**

(Only for new nonprovisional applications under 37 C.F.R. § 1.53(b))

Attorney Docket No. MTI-31072

First Inventor or Application Identifier Ole Bentz

Title See 1 in Addendum

Express Mail Label No. EL700460876US

**APPLICATION ELEMENTS**

See MPEP chapter 600 concerning utility patent application contents.

ADDRESS TO:

Assistant Commissioner for Patents  
Box Patent Application  
Washington, DC 20231

1. ☒ \* Fee Transmittal Form (e.g., PTO/SB/17)  
(Submit an original and a duplicate for fee processing)
2. ☒ Specification [Total Pages 19] 1  
(preferred arrangement set forth below)  
- Descriptive title of the Invention  
- Cross References to Related Applications  
- Statement Regarding Fed sponsored R & D  
- Reference to Microfiche Appendix  
- Background of the Invention  
- Brief Summary of the Invention  
- Brief Description of the Drawings (if filed)  
- Detailed Description  
- Claim(s)  
- Abstract of the Disclosure
3. ☒ Drawing(s) (35 U.S.C. 113) [Total Sheets 5] 1
4. Oath or Declaration [Total Pages 3] 1  
a. ☒ Newly executed (original or copy)  
b. ☐ Copy from a prior application (37 C.F.R. § 1.63(d))  
(for continuation/divisional with Box 16 completed)  
i. ☐ **DELETION OF INVENTOR(S)**  
Signed statement attached deleting  
inventor(s) named in the prior application,  
see 37 C.F.R. §§ 1.63(d)(2) and 1.33(b).

5. ☐ Microfiche Computer Program (Appendix)
6. Nucleotide and/or Amino Acid Sequence Submission  
(if applicable, all necessary)  
a. ☐ Computer Readable Copy  
b. ☐ Paper Copy (identical to computer copy)  
c. ☐ Statement verifying identity of above copies

**ACCOMPANYING APPLICATION PARTS**

7. ☐ Assignment Papers (cover sheet & document(s))
8. ☐ 37 C.F.R. § 3.73(b) Statement of Attorney  
(when there is an assignee) ☐ Power of Attorney
9. ☐ English Translation Document (if applicable)
10. ☒ Information Disclosure Statement (IDS)/PTO-1449 ☒ Copies of IDS Citations
11. ☐ Preliminary Amendment
12. ☒ Return Receipt Postcard (MPEP 503)  
(Should be specifically itemized)
13. ☐ \* Small Entity Statement(s) filed in prior application  
(PTO/SB/09-12) ☐ Status still proper and desired
14. ☐ Certified Copy of Priority Document(s)  
(if foreign priority is claimed)
15. ☐ Other: \_\_\_\_\_

\* NOTE FOR ITEMS 1 & 13 IN ORDER TO BE ENTITLED TO PAY SMALL ENTITY FEES, A SMALL ENTITY STATEMENT IS REQUIRED (37 C.F.R. § 1.27), EXCEPT IF ONE FILED IN A PRIOR APPLICATION IS RELIED UPON (37 C.F.R. § 1.28).

**16. If a CONTINUING APPLICATION, check appropriate box, and supply the requisite information below and in a preliminary amendment:**☐ Continuation ☐ Divisional ☐ Continuation-in-part (CIP)

of prior application No: \_\_\_\_\_ / \_\_\_\_\_

Prior application information: Examiner \_\_\_\_\_

Group / Art Unit: \_\_\_\_\_

For CONTINUATION or DIVISIONAL APPS only: The entire disclosure of the prior application, from which an oath or declaration is supplied under Box 4b, is considered a part of the disclosure of the accompanying continuation or divisional application and is hereby incorporated by reference. The incorporation can only be relied upon when a portion has been inadvertently omitted from the submitted application parts.

**17. CORRESPONDENCE ADDRESS**☒ Customer Number or Bar Code Label

022202

or ☐ Correspondence address below

(Insert Customer No. or Attach bar code label here)

Name				
Address				
City	State	Zip Code		
Country	Telephone	Fax		

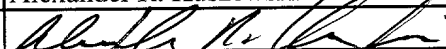
Name (Print/Type)

Alexander R. Kuszewski

Registration No. (Attorney/Agent)

41920

Signature



Date

8/30/2000

Burden Hour Statement. This form is estimated to take 0.2 hours to complete. Time will vary depending upon the needs of the individual case. Any comments on the amount of time you are required to complete this form should be sent to the Chief Information Officer, Patent and Trademark Office, Washington, DC 20231. DO NOT SEND FEES OR COMPLETED FORMS TO THIS ADDRESS. SEND TO: Assistant Commissioner for Patents, Box Patent Application, Washington, DC 20231.

[illegible]

- [illegible]

# OVERFLOW DETECTION AND CLAMPING WITH PARALLEL OPERAND PROCESSING FOR FIXED-POINT MULTIPLIERS

## BACKGROUND OF THE INVENTION

5           The present invention relates generally to the multiplication and clamping prediction of fixed-point multipliers. More particularly, the invention relates to a method and apparatus for increasing the speed of fixed-point data paths that involve multiplication of operands and parallel overflow detection and clamping based upon the magnitude of those operands.

10           Electrical circuits are routinely employed to perform arithmetic operations of operands represented by logical representations. Generally, it is desirable for arithmetic circuitry, and in particular multiplication circuitry, to have the fewest number of bits in order to perform the required calculations. Minimization of the required number of bits facilitates speed of the calculating circuit. However, counter-balancing the desire to have a limited number of bits to perform a given calculation is to avoid answer overflow. Overflow, or the situation in which  
15           an answer will exceed the number of bits designed for the answer, is not acceptable, as valuable valid data may be lost in performing the calculations. In circuits that perform calculations with a number of bits that may yield answers that overflow the set number of bits, clamping can be used to ensure that a result that overflows is clamped to a given acceptable value. Normally, the largest magnitude positive or negative representable number is  
20           employed as the overflow value. While the employment of a clamping operation is not always desirable, it is generally considered to be better than an overflow which may cause wrapping or undesired bits stored in the particular multiplication circuitry.

25           Typically, when clamping is desired, it is performed in a sequential fashion. In other words, the arithmetic operation is performed first, and when the result is available, it is then analyzed for overflow. If clamping is required, a clamping value replaces the computed value.

          Referring now to Fig. 1, a prior system illustrates the serial processing in which the clamping analysis follows multiplication of particular operands. In this instance, operand 1

(reference numeral 2) and operand 2 (reference numeral 4) are input into an arithmetic operator 6. The output 8 from the arithmetic operator 6 is then directed into the overflow detection scheme 10. Any delays between arithmetic operator 6 and the overflow detection 10 are compounded, with each of their respective delays adding to the overall circuit delay. The clamp value 12 as determined by the overflow detection 10 is input along with the result 14 of the arithmetic operation 6, and a preset clamp value 16 into multiplexer 18. Multiplexer 18 selects the operation result 14 or the preset clamp value 16 in the instance of overflow detection based upon the logical level of clamp bit 12. The result is then output into register 20.

A major problem with sequential operation is that any delays, for both the arithmetic operation and the overflow analysis, tend to compound and therefore yield a relatively slow circuit to the extent that all delays are combined and added together to determine the total circuit delay. In any arithmetic operation, multiplication delays tend to be the largest. Therefore, there exists a need to implement a circuit and method in which arithmetic operations such as multiplication may be performed along with overflow detection. The result is a circuit that eliminates the compounded delay made up of the accumulation of each individual delay associated with sequential overflow analysis and operand processing.

### **SUMMARY OF THE INVENTION**

The present invention provides overflow detection and clamping in parallel with multiplication of fixed-point multiplier operands that overcome the aforementioned problems, and provides a faster circuit than would otherwise be available from serial clamping analysis and arithmetic operation.

In accordance with one aspect of the invention, a method of detecting overflow in a clamping circuit includes inputting a first operand having a fixed-point format (i.e. represented by a fixed number of bits) into the clamping circuit and inputting a second operand having a second fixed-point format into the clamping circuit. A product overflow

output is determined based upon the first and second fixed-point format. The method includes predicting whether multiplication of the first operand with the second operand yields a result that exceeds the product overflow output, and performing at least partially the multiplication of the first and second operands. The determining step occurs substantially in parallel with the performing step.

In accordance with another aspect of the invention, a method of clamp detection is disclosed, and includes inputting a first and second operand to both a multiplier and an overflow detection circuit. The method includes multiplying the first and second operands to generate a result not to exceed a pre-determined number of bits, and determining an initial clamping predictor bit based upon the first operand and the second operand. The initial clamping predictor bit is logically ORed with a most significant bit of the result to produce a final clamping predictor bit.

Again, the multiplying and determining steps occur substantially in parallel.

In yet another aspect of the invention, a multiplication overflow detection circuit is disclosed. The circuit includes multiplication circuitry for at least partially multiplying a first and a second operand, overflow detection circuitry receiving the first and second operands that detects whether a result of the multiplication of the first and second operands exceed a maximum representable positive or negative value. The multiplication circuitry and the overflow detection circuitry operate substantially in parallel.

### **BRIEF DESCRIPTION OF THE DRAWINGS**

The drawings illustrate the best mode presently contemplated for carrying out the invention.

In the drawings:

Fig. 1 is a schematic of a prior detection overflow scheme showing serial operand operation and overflow detection.

Fig. 2 is a schematic of parallel operation of the arithmetic operator and the overflow detection in accordance with the present invention.

Fig. 3 is a table illustrating necessary information in determining whether to clamp multiplication of the two operands in accordance with one aspect of the present invention.

5 Fig. 4 is a graph illustrating the simple overflow predictor and the regions necessary for additional overflow prediction calculations.

Fig. 5 shows a schematic of another aspect of the present invention.

### **DETAILED DESCRIPTION OF THE PREFERRED EMBODIMENT**

10 Referring now to Fig. 2, a clamping system 22 in accordance with the present invention is illustrated. Operand 1 is loaded into register 24 and operand 2 is similarly loaded into register 26. Registers 24, 26 are preferably flip-flops, but any suitable register capable of storing the operands are contemplated. Also, the operands may be passed directly from other arithmetic operators or other related logic, if suitable for the application. Operand 1 and  
15 operand 2 are loaded in a fixed-point format.

In fixed-point arithmetic, numbers are represented by zero or one sign bit, zero or more integer bits, and zero or more fractional bits. The integer and fractional bits can be a magnitude, a 1's complement, or a 2's complement value. The most common case is the 2's complement case. A short-hand description is used to denote how many bits there are in each  
20 category. This description is typically: <sign-bits>.<integer-bits>.<fractional-bits> or *s.i.f*. For a signed number with 4 integer bits and 5 fractional bits the notation would be 1.4.5. For an unsigned number with 6 integer bits and 3 fractional bits the notation would be 0.6.3, and often the leading zero is omitted to yield 6.3.

For a given fixed-point representation *s.i.f* there are associated numerical properties,  
25 (and assuming that the 2's complement is used):

Largest positive number representable:  $2^i - 2^{-f}$

Largest (magnitude) negative number representable:  $-2^i$   
Smallest positive number representable:  $2^{-f}$   
Smallest (magnitude) negative number representable:  $-2^{-f}$

5 When fixed-point numbers are used in arithmetic operations such as additions, subtractions, and multiplications, the results generally require more bits to avoid overflow.

The rules for the two types of operations are:

1. Addition and subtraction:

- 10
- If both inputs are of the same format  $s.i.f$  then the output requires  $s.(i+1).f$  to avoid overflow.
  - If the inputs are of different formats  $s.i1.f1$  and  $s.i2.f2$ , then choose  $i = \max(i1, i2)$  and  $f = \max(f1, f2)$ . Then the output requires  $s.(i+1).f$  bits to avoid overflow.

15

2. Multiplication:

- If the inputs are of the same format  $s.i.f$  then the output requires  $s.(2*i+1).(2*f)$  to avoid overflow.
  - If the inputs are of different formats  $s.i1.f1$  and  $s.i2.f2$ , then the output requires  $s.(i1+i2+1).(f1+f2)$  to avoid overflow.
- 20

25

Since multiplication most often produces the need for overflow detection and clamping, it is preferred that the present invention be utilized in multiplication of operand 1 and operand 2, with each operand in a fixed-point format. As Fig. 2 demonstrates, operand 1 and operand 2 are both supplied to the overflow detection 28 as well as the arithmetic operator or multiplier 30. The multiplier 30 takes operand 1 and operand 2 and determines at least a partial product of the binary operands. It is important to note that the multiplier does

not have to be full precision. It need only be twice the precision that is desired to be represented in the final result. For example, if it is desired to multiply two operands, each having eight bits, when the multiplication of eight bits and eight bits occurs, the answer will have sixteen bits of output. However, if it is only desired to have eight bits to come out in the final output, when the multiplication could actually generate up to sixteen bits, the multiplier will be used to generate nine bits. The nine bits represent twice the desired output range. Therefore, in the above-identified example, the answer will have a nine-bit result instead of the full sixteen-bit result. The final seven bits of the multiplication are not necessarily calculated.

Operand 1 and operand 2 are also presented to the overflow detection logic. The overflow detection 28 occurs in parallel with the multiplier 30 operation. It is the substantially simultaneous processing of the overflow detection and the multiplication that produces the efficiencies in processing time for the circuit. Also, it is the fact that the full product need not necessarily be calculated that results in further efficiencies in processing time.

The overflow detection circuit 28 considers the two input operands and predicts whether or not they will cause the product to overflow. The overflow prediction circuit 28 predicts when the product is going to be greater than or equal to two times the maximum desired value. This handles most of the clamping cases, but it is not exact. To get exact clamping, the product is computed to the  $(io+1)$ 'th integer bit, and the most significant bit of the product 37 is used to finally determine whether or not to clamp.

The fixed-point format of the two inputs and the desired output must be known. These formats are denoted as:

Operand1: *s.i1.f1*  
Operand2: *s.i2.f2*  
Output: *s.io.fo*



The product of the two input operands will require  $s.(i1+i2+1).(f1+f2)$  to avoid overflow. The assumption is that  $io < i1+i2+1$ , (because, if  $io > i1+i2+1$  then an overflow can not occur).

## 5                    *Example*

Consider the simple case of two positive input operands that yield a positive product. The maximum positive number that can be represented by the desired output representation is:

10                     $2^{io} - 2^{-fo}$

Therefore, any combination of input operands that yields a product greater than or equal to the maximum value should be clamped to the maximum positive value:

15                    
$$\begin{array}{ll} \text{Output} = \text{op1} * \text{op2} & \text{if } (\text{op1} * \text{op2}) < 2^{io} - 2^{-fo} \\ \text{Output} = 2^{io} - 2^{-fo} & \text{if } (\text{op1} * \text{op2}) \geq 2^{io} - 2^{-fo} \end{array}$$

20                    The result 32 of multiplier 30 (which will in a preferred embodiment be a partial multiplication) is output from the multiplier 30. The detection overflow 28 outputs result 34 from its overflow detection circuitry, and as a result of a logical level of the clamp bit 34 it is determined whether clamping will occur. The most significant bit 37 (on line 36) of result 32 is logically ORed with clamp bit 34 through OR gate 38. Therefore, if any of clamp bit 34 or the most significant bit 37 is logically high, clamping will occur and a clamping signal 40 is output from OR gate 38. The clamping signal 40 is input into multiplexer 42. Result 32 of

25                    the multiplication is also input into multiplexer 42. Clamp value 44 is preferably hard-wired into multiplexer 42. The value of clamp value 44 is predetermined depending on the number of bits in operand 1 and operand 2. Preferably, the clamp value is the maximum representable positive or negative value. It is contemplated that in selecting a positive or negative value for

operand 1 or operand 2, the MSB (most significant bit) of each register 24, 26 could be exclusively ORed (XOR) together such that if the operands are of a different sign, the negative clamping value will be used for clamp value 44. Consequently, if both operands are of the same sign, the positive value for clamp value 44 will be used. Multiplexer 42 will  
 5 select either result 32 or clamp value 44 depending upon the logical level of clamp signal 40, and will output the selection into output register 46.

Referring now to Fig. 3, a binary representation of several scenarios are given.

Take the case of two 1.6.4 operands, multiply them and return the product clamped to a 1.6.4 number. The maximum representable value in the output is  $2^6 - 2^{-4} = 64 - 1/16 =$   
 10 63.9375. Some simple cases exist:

If  $op1 \geq 32$  and  $op2 \geq 2$  then clamp (45a)

If  $op1 \geq 16$  and  $op2 \geq 4$  then clamp (45b)

If  $op1 \geq 8$  and  $op2 \geq 8$  then clamp (45c)

15 If  $op1 \geq 4$  and  $op2 \geq 16$  then clamp (45d)

If  $op1 \geq 2$  and  $op2 \geq 32$  then clamp (45e)

The binary representation of these scenarios is given in Fig. 3.

20 It can be seen that the number of leading zeros in the integer portion of the operands is indicative of the magnitude of the operands, and by adding the number of leading zeros of both  $op1$  (47) and  $op2$  (49), there is a constant number of leading zeros 51. Therefore, the fixed-point format of the operands will determine the constant number of leading zeros to determine whether clamping occurs. Clamping must occur when:

25 If  $(\langle op1 \text{ leading zeros} \rangle + \langle op2 \text{ leading zeros} \rangle) \leq 4$  then clamp  
 Else don't clamp

This relatively simple predictor works for a substantial portion of products, but it is not completely accurate. If only this predictor were used, it would be possible to get products that could be as much as twice the desired max value, i.e.,

$$(\text{max product given simple predictor}) < 2 * (\text{desired max value})$$

5

Therefore, to get an accurate clamp predictor, the above simple predictor is used, and the multiplication generates a result that is of format s.(io+1).fo. Then, for positive operands, an overflow is detected by ORing together the MSB of the integer bits and the clamp-prediction.

10

Referring now to Fig. 4, a graph is shown showing the regions of products that are clamped by the initial clamp predictor. The broad region covered by the simple, initial clamp predictor is shown generally by the numeral 50. Regions 52 represent those products that require the io+1 integer bits of the product to get accurate overflow detection and therefore require more precise clamp prediction. Region 51 represents the region where clamping is not needed because the product does not exceed the desired number of bits. However, this graph is representative of only one particular set of operands (both positive). Other predictors will produce different data depending upon the signs of the operands.

15

### ***General Case***

20

The general case consists of three subcases depending on the signs of the input operands:

25

- Case 1: (op1 > 0 and op2 > 0)
- Case 2: (op1 < 1 and op2 < 1)
- Case 3: (op1 > 0 and op2 < 1) or (op1 < 0 and op2 > 0)

Each of these cases will be examined in turn.

***Case 1:*** Both operands are positive

The simple clamp predictor is:

If the sum of the input operands' *leading zeros* is less than or equal to

$$\begin{aligned} & (i1 - io) + (i2 - io) + (io - 2) \\ & = i1 + i2 - io - 2 \end{aligned}$$

then the circuit must clamp.

The accurate clamp prediction must use the  $(io + 1)$  bit of the product, i.e., the product must be computed at least to  $(io + 1)$  integer bits. This bit is ORed logically with the simple clamp predictor to yield the accurate clamp determination.

**Case 2:** Both operands are negative

When both operands are negative, their product is positive. Therefore, if an overflow case exists we clamp to the same value as mentioned before, namely:

$$2^{io} - 2^{-fo}$$

When both operands are negative the simple clamp predictor must count *leading ones* in the input operands. If the sum of the input operands' leading ones is less than or equal to

$$\begin{aligned} & (i1 - io) + (i2 - io) + (io - 1) \\ & = i1 + i2 - io - 1 \end{aligned}$$

then the circuit must clamp.

However, there is an additional case, when both operands have only zeros after the leading ones (e.g., 111100.0000) then the simple clamp predictor should also clamp. This is like counting leading zeros on a bit-reversed version of each input. If the trailing zeros plus

the leading ones equal (i+f) for both inputs then clamp. As before, the accurate clamp prediction must use the (io + 1) bit of the product, i.e., the product must be computed at least to (io + 1) integer bits. This bit is ORed logically with the simple clamp predictor to yield the accurate clamp determination.

5

**Case 3: Only one of the operands is negative**

When only one of the input operands is negative, their product will be negative. Therefore, if an overflow case exists we clamp to the largest (magnitude) negative value, namely,  $-2^{i_o}$ .

10

In this case, the simple clamp predictor must count *leading ones* for the negative input, and *leading zeros* for the positive input. Then, if the sum of the inputs' leading ones and leading zeros is less than or equal to

15

$$\begin{aligned} & (i_1 - i_o) + (i_2 - i_o) + (i_o - 2) \\ & = i_1 + i_2 - i_o - 2 \end{aligned}$$

then the circuit must clamp.

20

As before, the accurate clamp prediction must use the (io + 1) bit of the product, i.e., the product must be computed at least to (io + 1) integer bits. This bit is logically inverted, then ORed logically with the simple clamp predictor to yield the accurate clamp determination.

25

Referring now to Fig. 5, another embodiment of the present invention is shown. In this embodiment, the clamp bit may be stored in a clamp bit register 60 and the result of the multiplier operation may be stored in multiplier result register 62. In this embodiment, after

one of the result 62 and the clamp value 64 is selected, other logic 68 is introduced to process the output 66 as part of a pipelining stage. The pipelining stage enables further processing of a partial result if the computation is not completed in a single clock cycle. The clamp prediction along with a partial result is then saved for the following clock cycle where completion of the clamping prediction will occur in the second clock cycle.

The initial clamp predictor may be utilized as soon as the two operands are available. The final clamp predictor must occur after the result of the multiplier has completed its partial computation to the appropriate necessary bit. Therefore, the simple clamp prediction occurs at the same time as the multiplier as manipulating operand 1 and operand 2.

Although two operands are shown, it is contemplated by the present invention that any number of operands may be used as inputs to the multiplier operation. In addition, the multiplication may occur in several stages with other multiplications coming before or after in a similar manner. The cloud of logic represents other multiplications, other additions or other logic operations on the result 66 of the multiplexer.

The present invention has been described in terms of the preferred embodiment, and it is recognized that equivalents, alternatives, and modifications, aside from those expressly stated, are possible and within the scope of the appending claims.

## CLAIMS

1. A method of detecting overflow in a clamping circuit comprising the steps of:  
inputting a first operand having a first fixed-point format into the clamping  
5 circuit;  
inputting a second operand having a second fixed-point format into the  
clamping circuit;  
determining an overflow output based upon the first and second fixed-point  
format and predicting whether an arithmetic operation of the first operand with the second  
10 operand will yield a result that exceeds the overflow output; and  
performing at least partially the arithmetic operation of the first and second  
operands;  
wherein the determining and predicting step occurs substantially in parallel  
with the performing step.  
15
2. A method of detecting overflow in a clamping circuit comprising the steps of:  
inputting a first operand having a first fixed-point format into the clamping  
circuit;  
inputting a second operand having a second fixed-point format into the  
20 clamping circuit;  
determining a product overflow output based upon the first and second fixed-  
point format and predicting whether multiplication of the first operand with the second  
operand yields a result that exceeds the product overflow output; and  
performing at least partially the multiplication of the first and second operands;  
25 wherein the determining and predicting step occurs substantially in parallel  
with the performing step.
3. A method of clamping fixed-point multipliers:  
providing a first operand in a first fixed-point format;  
30 providing a second operand in a second fixed-point format;

at least partially multiplying the first operand with the second operand to produce an operation result;

determining whether the operation result will exceed a representable value;

determining a clamping value based on the first fixed-point format of the first operand and the second fixed-point format of the second operand; and

substituting the operation result with the clamping value when it is determined that the operation result will exceed the representable value;

wherein the multiplying step and determining whether the operation result will exceed the representable value step occur substantially in parallel.

10

4. A method of clamping fixed-point multipliers:

providing a first and second input operand;

determining a desired number of output bits;

where any of the first and second input operands are positive, counting a number of leading logical zeros in the positive operand;

where any of the first and second input operands are negative, counting a number of leading logical ones in the negative operands;

summing the number of leading logical zeros of the positive input operands with the number of leading logical ones in the negative input operands;

determining a clamping decision based on the summing step to yield a simple clamp predictor representative of the clamping decision;

computing a product of the first operand and the second operand such that the product has the desired number of output bits plus one additional bit;

logically ORing the simple clamp predictor with a most significant bit of the product.

5. The method of claim 4 wherein the computing step and determining the clamping decision to yield the simple clamp predictor step occur substantially in parallel.

6. A method of processing multiplier data paths comprising the steps of:



performing at least a partial multiplication of a plurality of operands, each having a fixed-point format;

determining whether the at least partial multiplication of the operands produces a product that will exceed a pre-determined limit based upon the fixed-point format of each of the operands; and

wherein the performance step and the determining step occur substantially in parallel.

7. A method of clamp detection comprising the steps of:

inputting a first and a second operand to both a multiplier and an overflow detection circuit;

multiplying the first and second operands to generate a result not to exceed a pre-determined number of bits;

determining an initial clamping predictor bit based upon the first operand and the second operand; and

logically ORing the initial clamping predictor bit and a most significant bit of the result to produce a final clamping predictor bit;

wherein the multiplying and determining steps occur substantially in parallel.

8. The method of claim 7 wherein the first and second operands are in a fixed-point format.

9. The method of claim 7 wherein the most significant bit of the result is logically inverted prior to the logically ORing step.

10. The method of claim 8 wherein the step of determining the initial clamping predictor bit includes determining a number of logical zeros in each of the operands and summing the number of logical zeros to determine whether the sum exceeds a pre-determined number to determine the initial clamping predictor bit.



a multiplier for at least partially multiplying the first and second operands and generating a result not to exceed a pre-determined number of bits;

a clamp bit register for storing the clamp bit from the overflow detection circuitry; and

5 a result register connected to the multiplier for storing the result of the multiplication of the first and second operands;

wherein the overflow detection circuitry and the multiplier operate substantially in parallel.

10 16. The overflow detection circuit of claim 15 further comprising:

a multiplexer comprising:

a clamp value input for receiving a clamp value to be output when clamping occurs;

15 a clamp bit register input connected to the clamp bit register for receiving the clamp bit;

a result register input connected to the result register for receiving the result of the multiplication of the first and second operands; and

an output;

20 wherein the multiplexer selects one of the clamp value register input and the result register input based upon a logical level of the clamp bit register in order to make the selected input the output of the multiplexer.

17. The overflow detection circuit of claim 15 wherein the clamp bit input is logically ORed with a most significant bit of the result stored in the result register.

25

18. The overflow detection circuit of claim 15 wherein the registers are flip-flops.

19. The overflow detection circuit of claim 15 wherein the first and second registers store the first and second operands in a fixed-point format.



Parameter	Value	Unit
Initial concentration	1.0	g/L
Initial pH	7.0	
Temperature	25	°C
Time	0, 1, 2, 4, 8, 16, 32, 64, 128, 256, 512, 1024, 2048, 4096, 8192, 16384, 32768, 65536, 131072, 262144, 524288, 1048576, 2097152, 4194304, 8388608, 16777216, 33554432, 67108864, 134217728, 268435456, 536870912, 1073741824, 2147483648, 4294967296, 8589934592, 17179869184, 34359738368, 68719476736, 137438953472, 274877906944, 549755813888, 1099511627776, 2199023255552, 4398046511104, 8796093022208, 17592186044416, 35184372088832, 70368744177664, 140737488355328, 281474976710656, 562949953421312, 1125899906842624, 2251799813685248, 4503599627370496, 9007199254740992, 18014398509481984, 36028797018963968, 72057594037927936, 144115188075855872, 288230376151711744, 576460752303423488, 1152921504606846976, 2305843009213693952, 4611686018427387904, 9223372036854775808, 18446744073709551616, 36893488147419103232, 73786976294838206464, 147573952589676412928, 295147905179352825856, 590295810358705651712, 1180591620717411303424, 2361183241434822606848, 4722366482869645213696, 9444732965739290427392, 18889465931478580854784, 37778931862957161709568, 75557863725914323419136, 151115727451828646838272, 302231454903657293676544, 604462909807314587353088, 1208925819614629174706176, 2417851639229258349412352, 4835703278458516698824704, 9671406556917033397649408, 19342813113834066795298816, 38685626227668133590597632, 77371252455336267181195264, 154742504910672534362390528, 309485009821345068724781056, 618970019642690137449562112, 1237940039285380274899124224, 2475880078570760549798248448, 4951760157141521099596496896, 9903520314283042199192993792, 19807040628566084398385987584, 39614081257132168796771975168, 79228162514264337593543950336, 158456325028528675187087900672, 316912650057057350374175801344, 633825300114114700748351602688, 1267650600228229401496703205376, 2535301200456458802993406410752, 5070602400912917605986812821504, 10141204801825835211973625643008, 20282409603651670423947251286016, 40564819207303340847894502572032, 81129638414606681695789005144064, 162259276829213363391578010288128, 324518553658426726783156020576256, 649037107316853453566312041152512, 1298074214633706907132624082305024, 2596148429267413814265248164610048, 5192296858534827628530496329220096, 10384593717069655257060992658440192, 20769187434139310514121985316880384, 41538374868278621028243970633760768, 83076749736557242056487941267521536, 166153499473114484112975882535043072, 332306998946228968225951765070086144, 664613997892457936451903530140172288, 1329227995784915872903807060280344576, 2658455991569831745807614120560689152, 5316911983139663491615228241121378304, 10633823966279326983230456482242756608, 21267647932558653966460912964485513216, 42535295865117307932921825928971026432, 85070591730234615865843651857942052864, 170141183460469231731687303715884105728, 340282366920938463463374607431768211456, 680564733841876926926749214863536422912, 1361129467683753853853498429727072845824, 2722258935367507707706996859454145691648, 5444517870735015415413993718908291383296, 10889035741470030830827987437816582766592, 21778071482940061661655974875633165533184, 43556142965880123323311949751266331066368, 87112285931760246646623899502532662132736, 174224571863520493293247799005065324265472, 348449143727040986586495598010130648530944, 696898287454081973172991196020261297061888, 1393796574908163946345982392040522594123776, 2787593149816327892691964784081045188247552, 5575186299632655785383929568162090376495104, 11150372599265311570767859136324180752990208, 22300745198530623141535718272648361505980416, 44601490397061246283071436545296723011960832, 89202980794122492566142873090593446023921664, 178405961588244985132285746181186892047843328, 356811923176489970264571492362373784095686656, 713623846352979940529142984724747568191373312, 1427247692705959881058285969449495136382746624, 2854495385411919762116571938898990272765493248, 5708990770823839524233143877797980545530986496, 11417981541647679048466287755595961091061972992, 2283596308329	

5

10

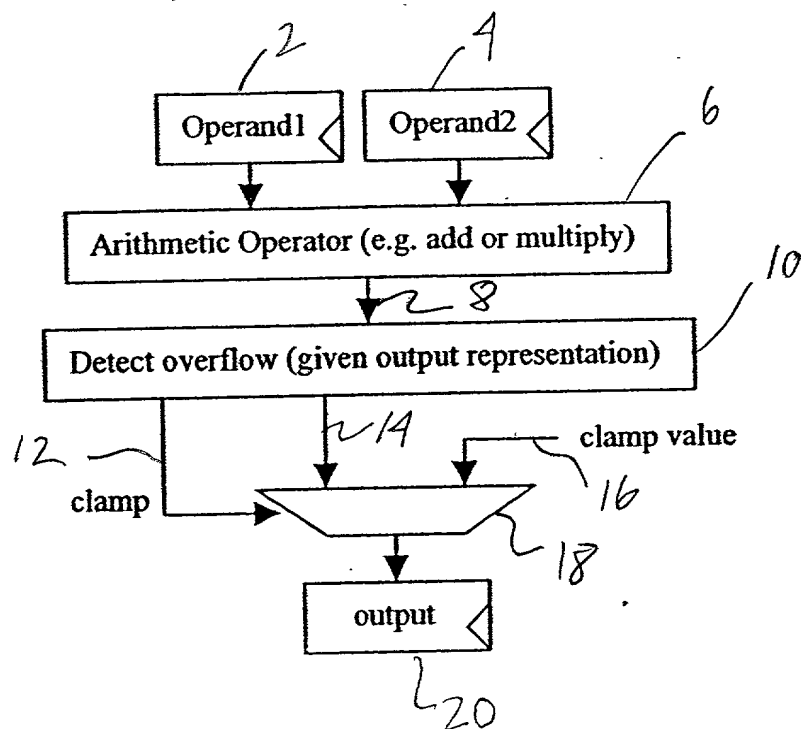
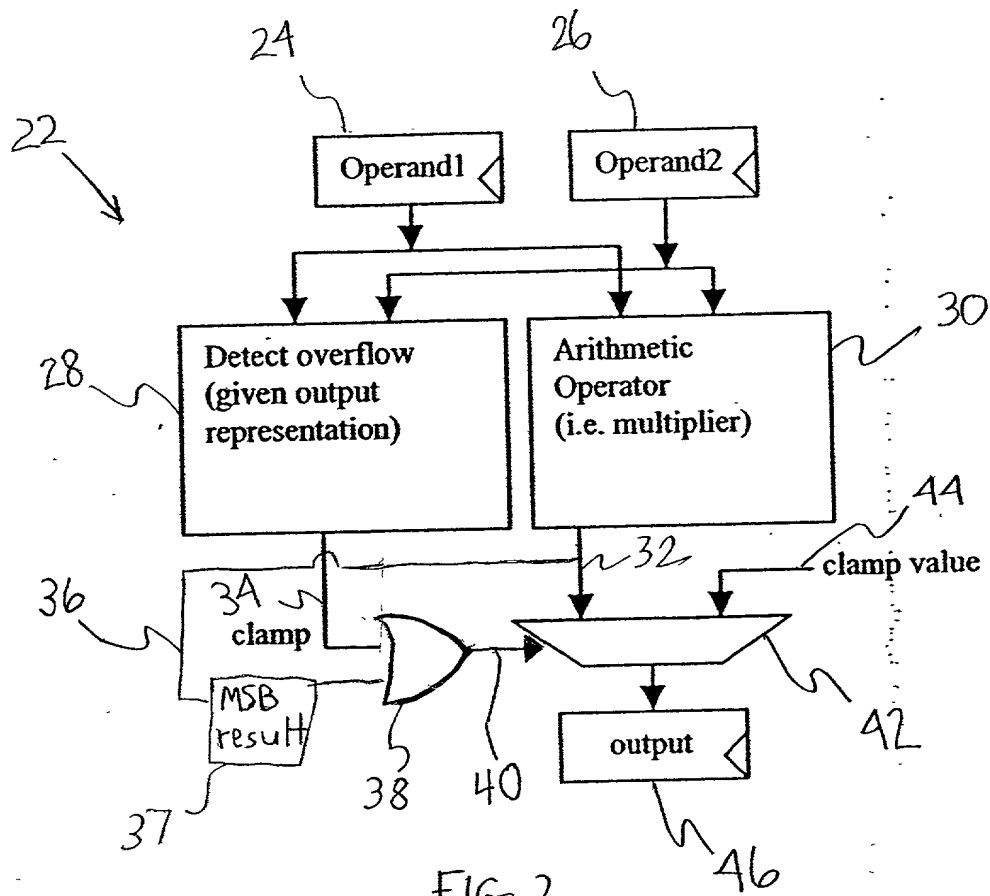
[illegible]

FIG. 1  
(PRIOR ART)



Clamp Case	Op1 >=	Op2 >=	Op1 leading zeros	Op2 leading zeros	Total leading zeros
1.	0.100000.0000	0.000010.0000	0	4	4
2.	0.010000.0000	0.000100.0000	1	3	4
3.	0.001000.0000	0.001000.0000	2	2	4
4.	0.000100.0000	0.010000.0000	3	1	4
5.	0.000010.0000	0.100000.0000	4	0	4

FIG. 3



### Clamp Curves

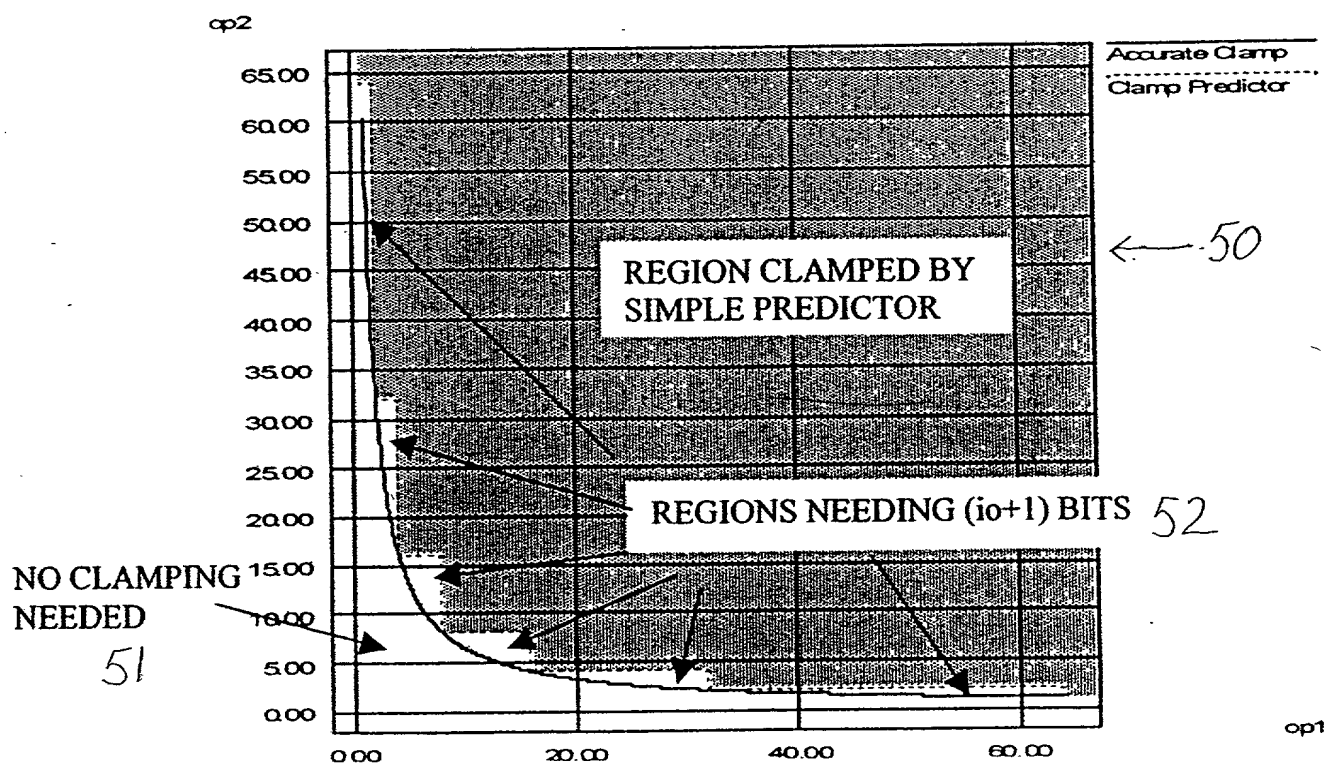


FIG. 4

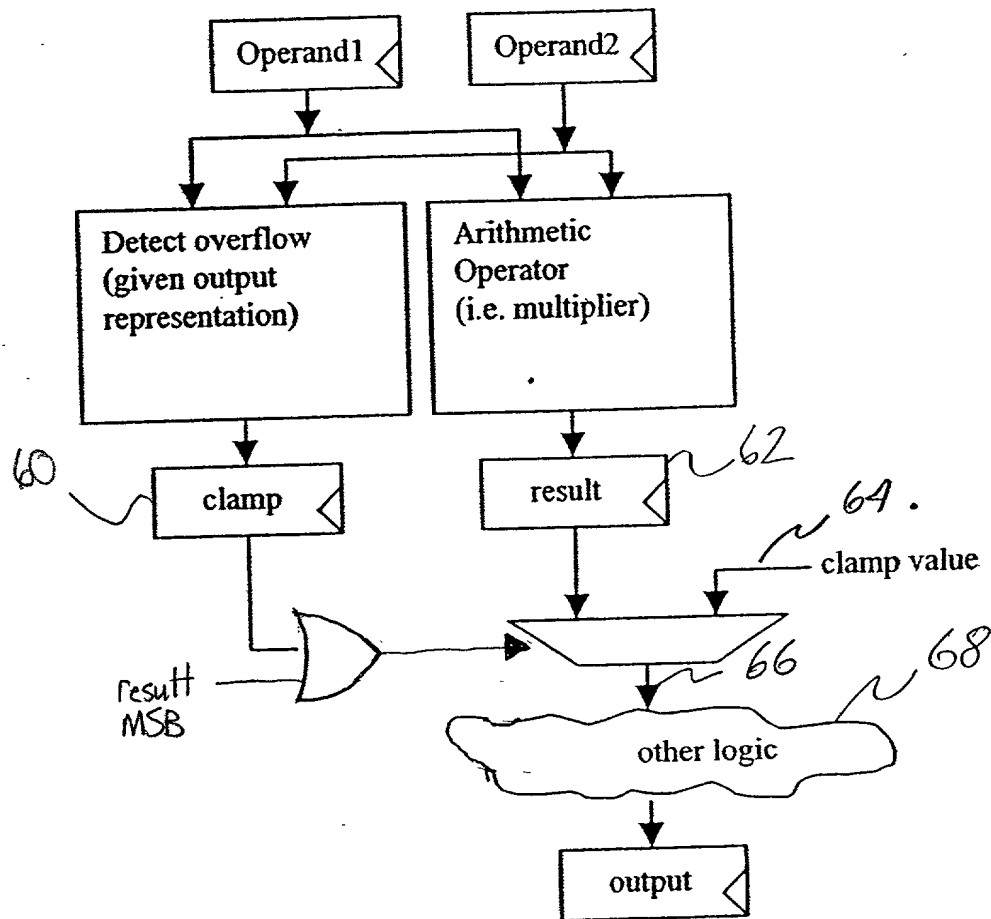
[illegible]

FIG. 5

Please type a plus sign (+) inside this box → ☐

PTO/SB/01 (12-97)  
Approved for use through 9/30/00 OMB 0851-0032  
Patent and Trademark Office; U.S. DEPARTMENT OF COMMERCE

Under the Paperwork Reduction Act of 1995, no persons are required to respond to a collection of information unless it contains a valid OMB control number.

<b>DECLARATION FOR UTILITY OR DESIGN PATENT APPLICATION (37 CFR 1.63)</b>  <input checked="" type="checkbox"/> Declaration Submitted with Initial Filing      OR <input type="checkbox"/> Declaration Submitted after Initial Filing (surcharge (37 CFR 1.16 (e)) required)	<b>Attorney Docket Number</b>	MTI-31072
	<b>First Named Inventor</b>	Bentz, Ole
	<b>COMPLETE IF KNOWN</b>	
	<b>Application Number</b>	
	<b>Filing Date</b>	
	<b>Group Art Unit</b>	
	<b>Examiner Name</b>	

As a below named inventor, I hereby declare that:

My residence, post office address, and citizenship are as stated below next to my name.

I believe I am the original, first and sole inventor (if only one name is listed below) or an original, first and joint inventor (if plural names are listed below) of the subject matter which is claimed and for which a patent is sought on the invention entitled:

**Overflow Detection and Clamping with Parallel Operand Processing for Fixed-Point Multipliers**

the specification of which (Title of the Invention)  
☒ is attached hereto  
OR  
☐ was filed on (MM/DD/YYYY)  as United States Application Number or PCT International Application Number  and was amended on (MM/DD/YYYY)  (if applicable).

I hereby state that I have reviewed and understand the contents of the above identified specification, including the claims, as amended by any amendment specifically referred to above.

I acknowledge the duty to disclose information which is material to patentability as defined in 37 CFR 1.56.

I hereby claim foreign priority benefits under 35 U.S.C. 119(a)-(d) or 356(b) of any foreign application(s) for patent or inventor's certificate, or 356(a) of any PCT international application which designated at least one country other than the United States of America, listed below and have also identified below, by checking the box, any foreign application for patent or inventor's certificate, or of any PCT international application having a filing date before that of the application on which priority is claimed.

Prior Foreign Application Number(s)	Country	Foreign Filing Date (MM/DD/YYYY)	Priority Not Claimed	Certified Copy Attached?	
				YES	NO
			<input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/>	<input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/>	<input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/>

☐ Additional foreign application numbers are listed on a supplemental priority data sheet PTO/SB/02B attached hereto.

I hereby claim the benefit under 35 U.S.C. 119(e) of any United States provisional application(s) listed below.

Application Number(s)	Filing Date (MM/DD/YYYY)

☐ Additional provisional application numbers are listed on a supplemental priority data sheet PTO/SB/02B attached hereto.

[Page 1 of 2]

Burden Hour Statement: This form is estimated to take 0.4 hours to complete. Time will vary depending upon the needs of the individual case. Any comments on the amount of time you are required to complete this form should be sent to the Chief Information Officer, Patent and Trademark Office, Washington, DC 20231. DO NOT SEND FEES OR COMPLETED FORMS TO THIS ADDRESS. SEND TO: Assistant Commissioner for Patents, Washington, DC 20231

Please type a plus sign (+) inside this box →

PTO/SB/01 (12-97)

Patent and Trademark Office, U.S. DEPARTMENT OF COMMERCE

Under the Paperwork Reduction Act of 1995, no persons are required to respond to a collection of information unless it contains a valid OMB control number.

## DECLARATION — Utility or Design Patent Application

I hereby claim the benefit under 35 U.S.C. 120 of any United States application(s), or 365(c) of any PCT international application designating the United States of America, listed below and, insofar as the subject matter of each of the claims of this application is not disclosed in the prior United States or PCT International application in the manner provided by the first paragraph of 35 U.S.C. 112, I acknowledge the duty to disclose information which is material to patentability as defined in 37 CFR 1.56 which became available between the filing date of the prior application and the national or PCT international filing date of this application.

U.S. Parent Application or PCT Parent Number	Parent Filing Date (MM/DD/YYYY)	Parent Patent Number (if applicable)

☐ Additional U.S. or PCT international application numbers are listed on a supplemental priority data sheet PTO/SB/02C attached hereto

As a named inventor, I hereby appoint the following registered practitioner(s) to prosecute this application and to transact all business in the Patent and Trademark Office connected therewith: ☐ Customer Number 002202 ☐ Blank

☐ Customer Number 022202

Place Customer  
Number Bar Code  
Label here

☒ Registered practitioner(s) name/registration number listed below

Name	Registration Number	Name	Registration Number
Gary R. Plotecher	27830	Alan E. Wagner	45188
Kristine M. Strodthoff	34259	Thomas J. Pienkos	P46992
Alexander R. Kuszewski	41920	Michael L. Lynch	30871

☒ Additional registered practitioner(s) named on supplemental Registered Practitioner Information sheet PTO/SB/02C attached hereto.

Direct all correspondence to: ☐ Customer Number  
or Bar Code Label


OR ☒ Correspondence address below

Name	Alexander R. Kuszewski, Esq.							
Address	Whyte Hirschboeck Dudek S.C.							
Address	111 E. Wisconsin Avenue, Suite 2100							
City	Milwaukee				State	WI	ZIP	53202
Country	USA		Telephone	414-273-2100			Fax	414-223-5000

I hereby declare that all statements made herein of my own knowledge are true and that all statements made on information and belief are believed to be true, and further that these statements were made with the knowledge that willful false statements and the like so made are punishable by fine or imprisonment, or both, under 18 U.S.C. 1001 and that such willful false statements may jeopardize the validity of the application or any patent issued thereon.

**Name of Sole or First Inventor:**

☐ A petition has been filed for this unsigned inventor

Given Name (first and middle [if any])					Family Name or Surname					
Ole					Bentz					
Inventor's Signature							Date		8/30/2000	
Residence City		Portland		State	OR	Country		USA	Citizenship	DK
Post Office Address		14604 NW Wendy Lane								
Post Office Address										
City		Portland		State	OR	ZIP		97229	Country	USA

☐ Additional inventors are being named on the 1 supplemental Additional Inventor(s) sheet(s) PTO/SB/02A attached hereto

